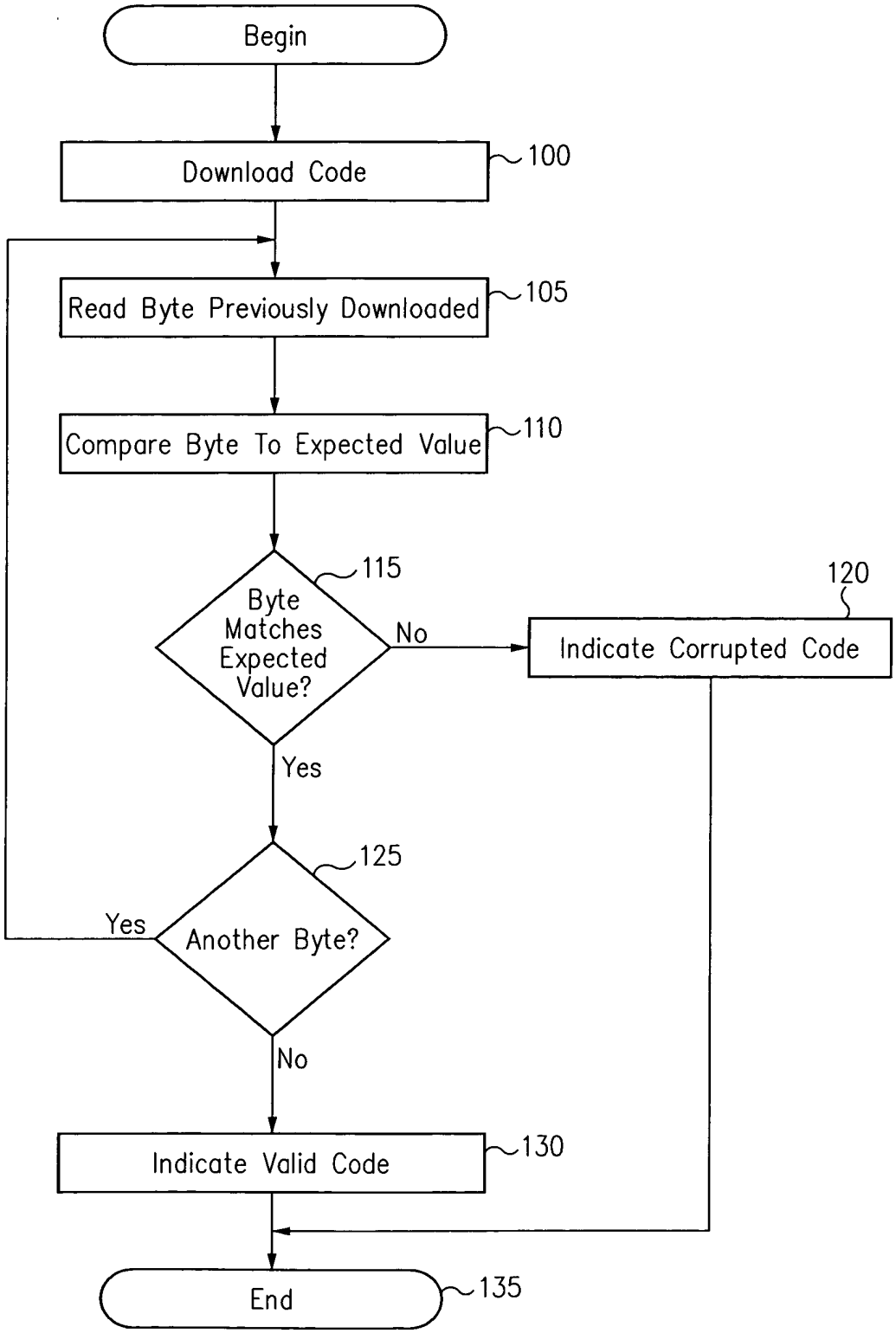

FIG. 1
(PRIOR ART)

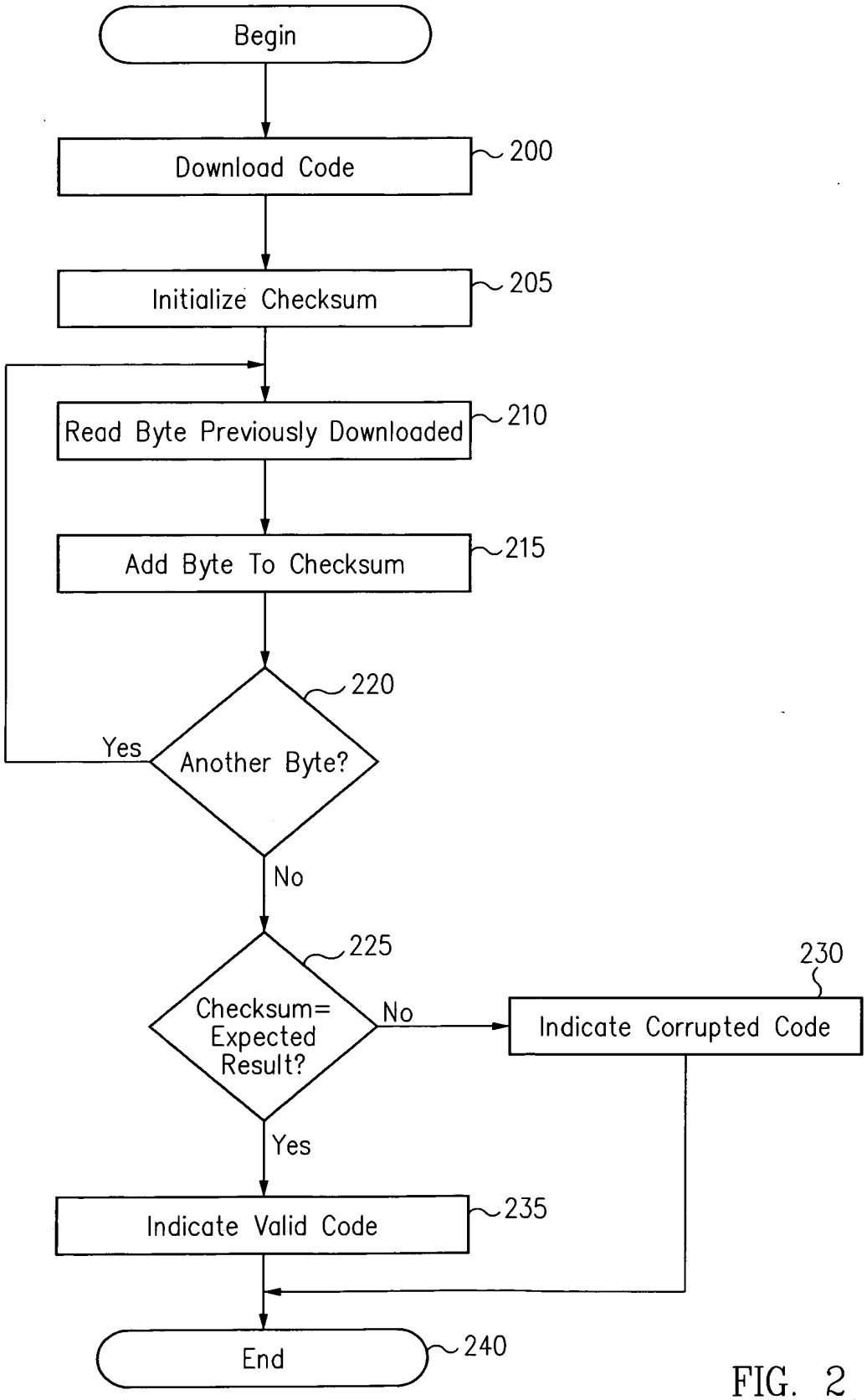

FIG. 2
(PRIOR ART)

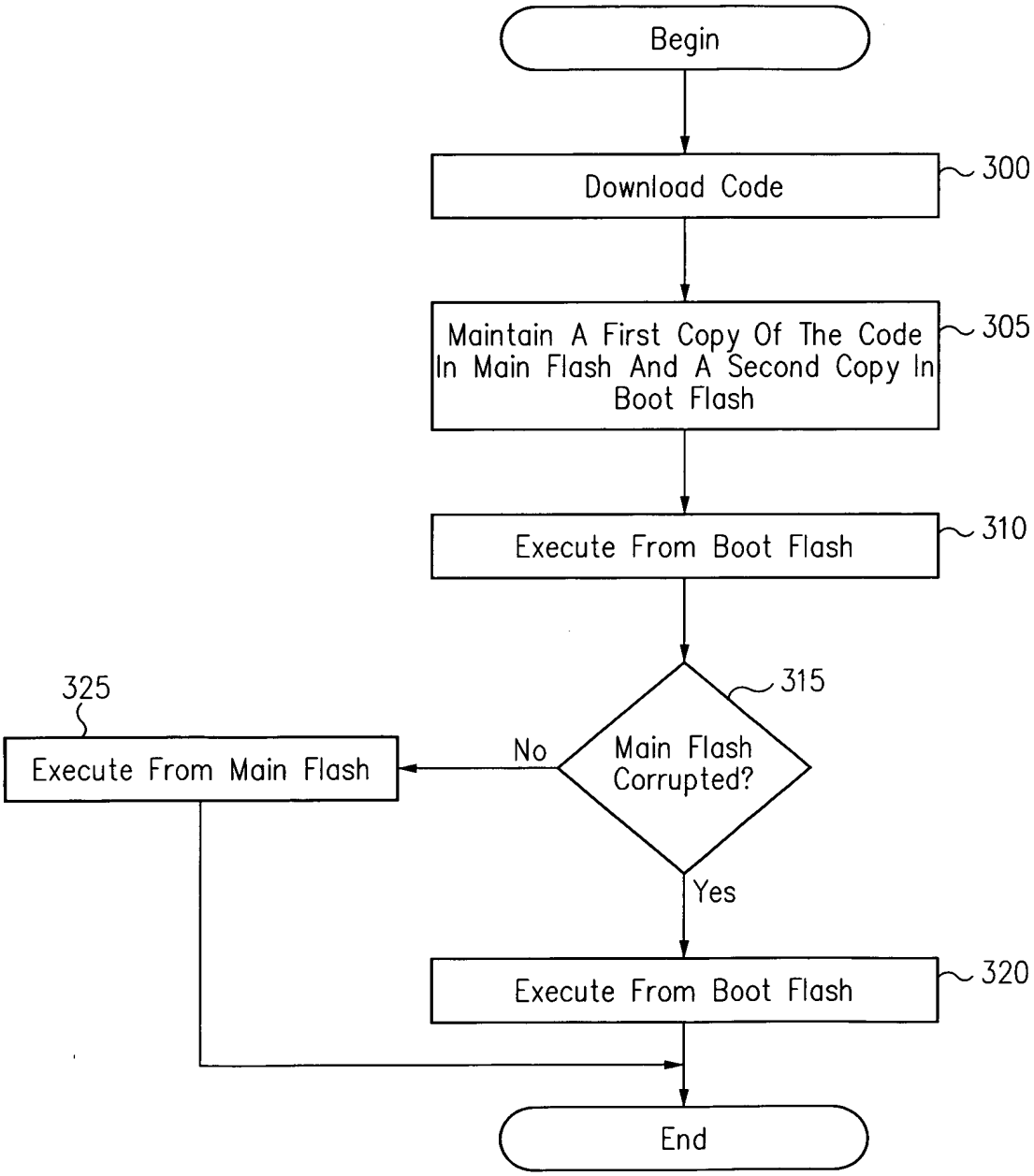

FIG. 3
(PRIOR ART)

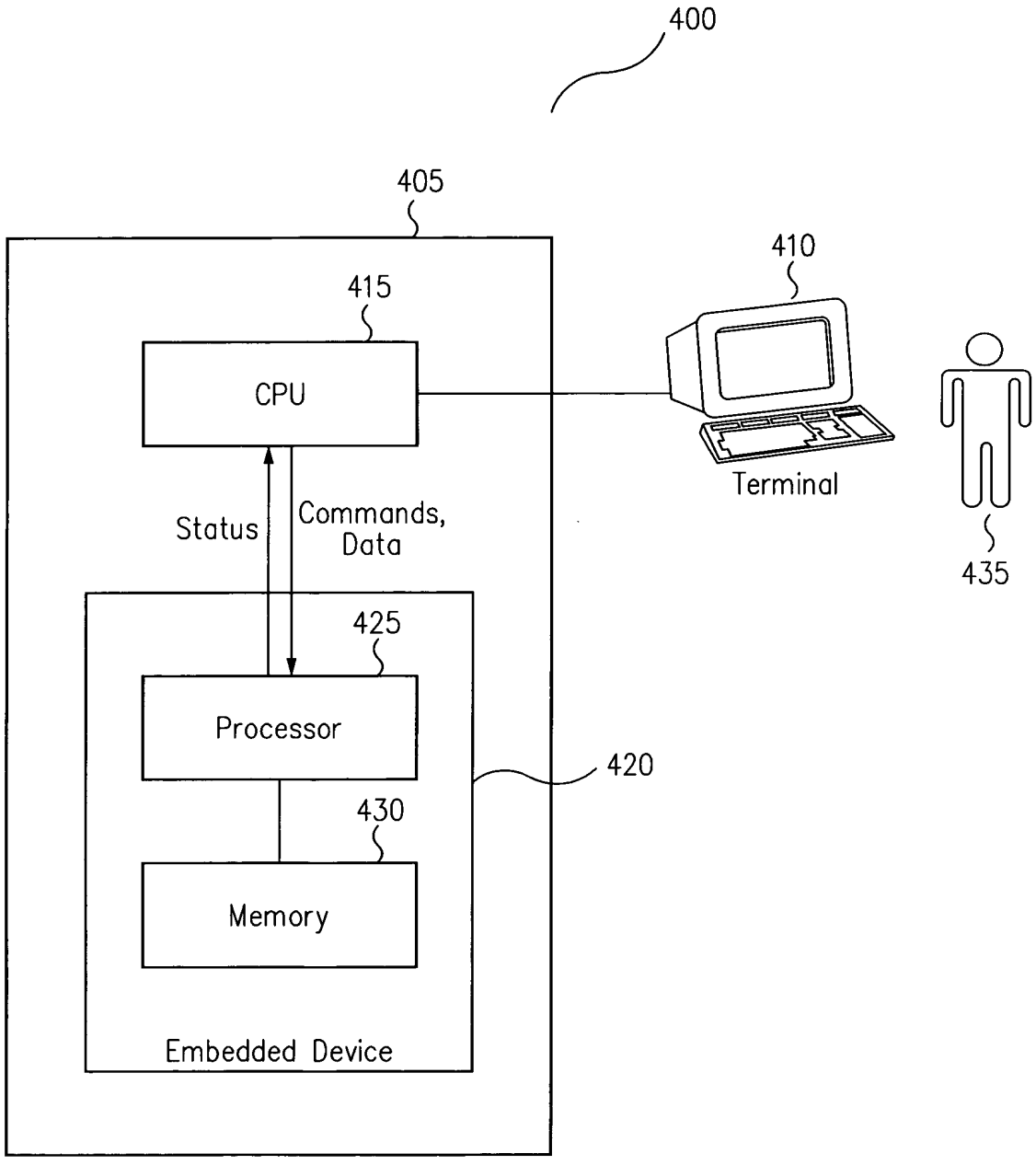
400
405
410
415
CPU
Terminal
435
Status
Commands,
Data
425
Processor
420
430
Memory
Embedded Device

FIG. 4

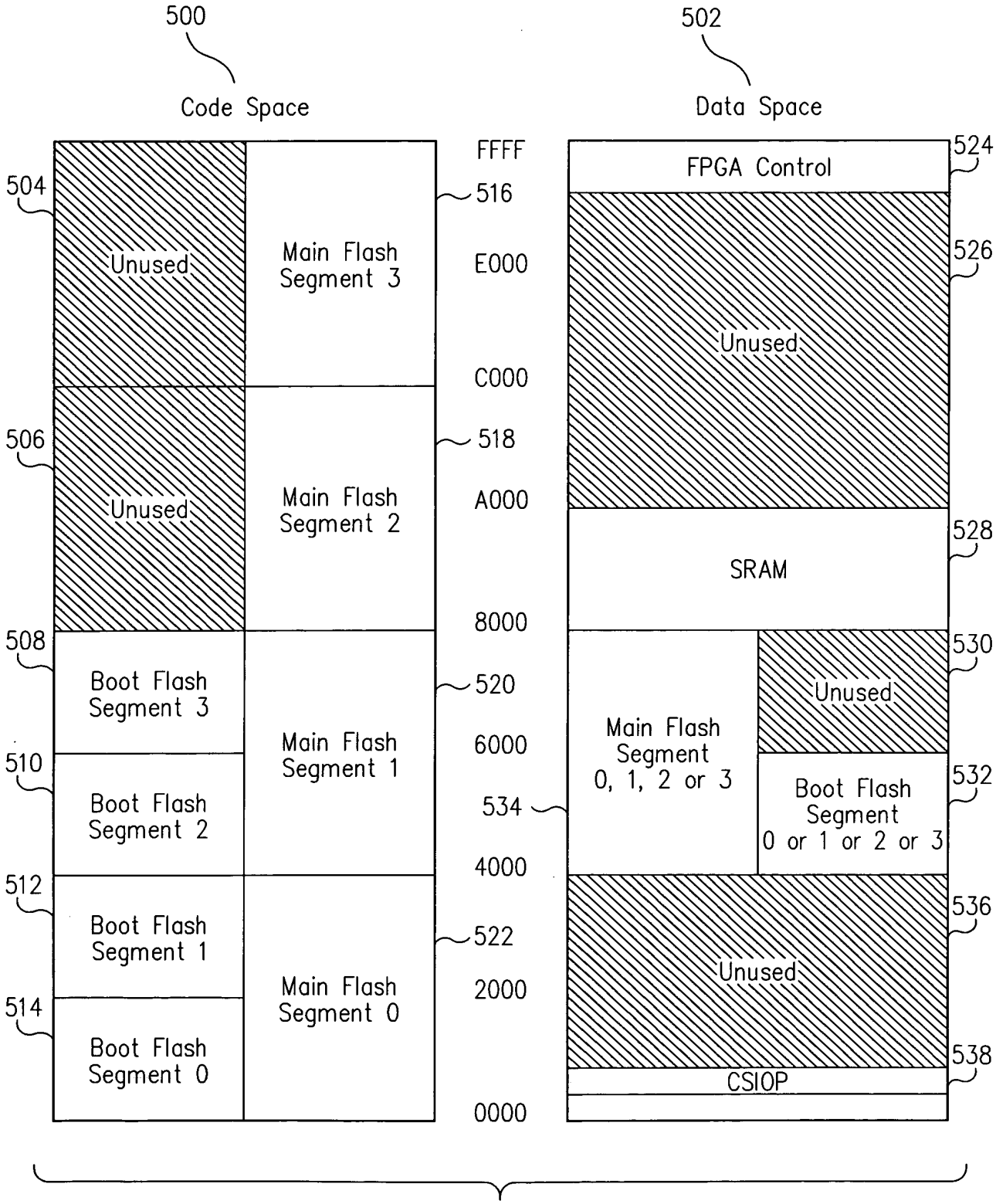
500
Code Space
502
Data Space
FFFF
E000
C000
A000
8000
6000
4000
2000
0000
504
Unused
Main Flash Segment 3
506
Unused
Main Flash Segment 2
508
Boot Flash Segment 3
510
Boot Flash Segment 2
512
Boot Flash Segment 1
514
Boot Flash Segment 0
Main Flash Segment 1
Main Flash Segment 0
516
518
520
522
534
524
FPGA Control
526
Unused
528
SRAM
530
Unused
Main Flash Segment 0, 1, 2 or 3
532
Boot Flash Segment 0 or 1 or 2 or 3
536
Unused
538
CSIOP

FIG. 5

600

Page Register

	Bit	Data	Note
605	0..2	Main Flash Segment Select	8 Segments Total
610	3..4	Boot Flash Segment Select	4 Segments Total
615	5	Flash Code Select	1: Main Flash Code 0: Boot Flash Code
620	6..7	Unused	

FIG. 6A

625

VM Register

	Bit	Data
630	0	SRAM Is Code
635	1	Boot Flash Is Code
640	2	Main Flash Is Code
645	3	Boot Flash Is Data
650	4	Main Flash Is Data
655	5..6	Unused
660	7	Enable PIO

FIG. 6B

Main Flash Segment 0 = (Device Space=Code) & (Flash Code Select=Main) & (address in 0000..3FFF)
OR (Device Space=Data) & (Main Flash Segment Select=0) & (address in 4000..7FFF))

Main Flash Segment 1 = (Device Space=Code) & (Flash Code Select=Main) & (address in 4000..7FFF)
OR (Device Space=Data) & (Main Flash Segment Select=1) & (address in 4000..7FFF))

Main Flash Segment 2 = (Device Space=Code) & (Flash Code Select=Main) & (address in 8000..BFFF)
OR (Device Space=Data) & (Main Flash Segment Select=2) & (address in 4000..7FFF))

Main Flash Segment 3 = (Device Space=Code) & (Flash Code Select=Main) & (address in C000..FFFF)
OR (Device Space=Data) & (Main Flash Segment Select=3) & (address in 4000..7FFF))

FIG. 7A

Boot Flash Segment 0 = (Device Space=Code) & (Flash Code Select=Boot) & (address in 0000..1FFF)
OR (Device Space=Data) & (Boot Flash Segment Select=0) & (address in 4000..5FFF))

Boot Flash Segment 1 = (Device Space=Code) & (Flash Code Select=Boot) & (address in 2000..3FFF)
OR (Device Space=Data) & (Boot Flash Segment Select=1) & (address in 4000..5FFF))

Boot Flash Segment 2 = (Device Space=Code) & (Flash Code Select=Boot) & (address in 4000..5FFF)
OR (Device Space=Data) & (Boot Flash Segment Select=2) & (address in 4000..5FFF))

Boot Flash Segment 3 = (Device Space=Code) & (Flash Code Select=Boot) & (address in 6000..7FFF)
OR (Device Space=Data) & (Boot Flash Segment Select=3) & (address in 4000..5FFF))

FIG. 7B

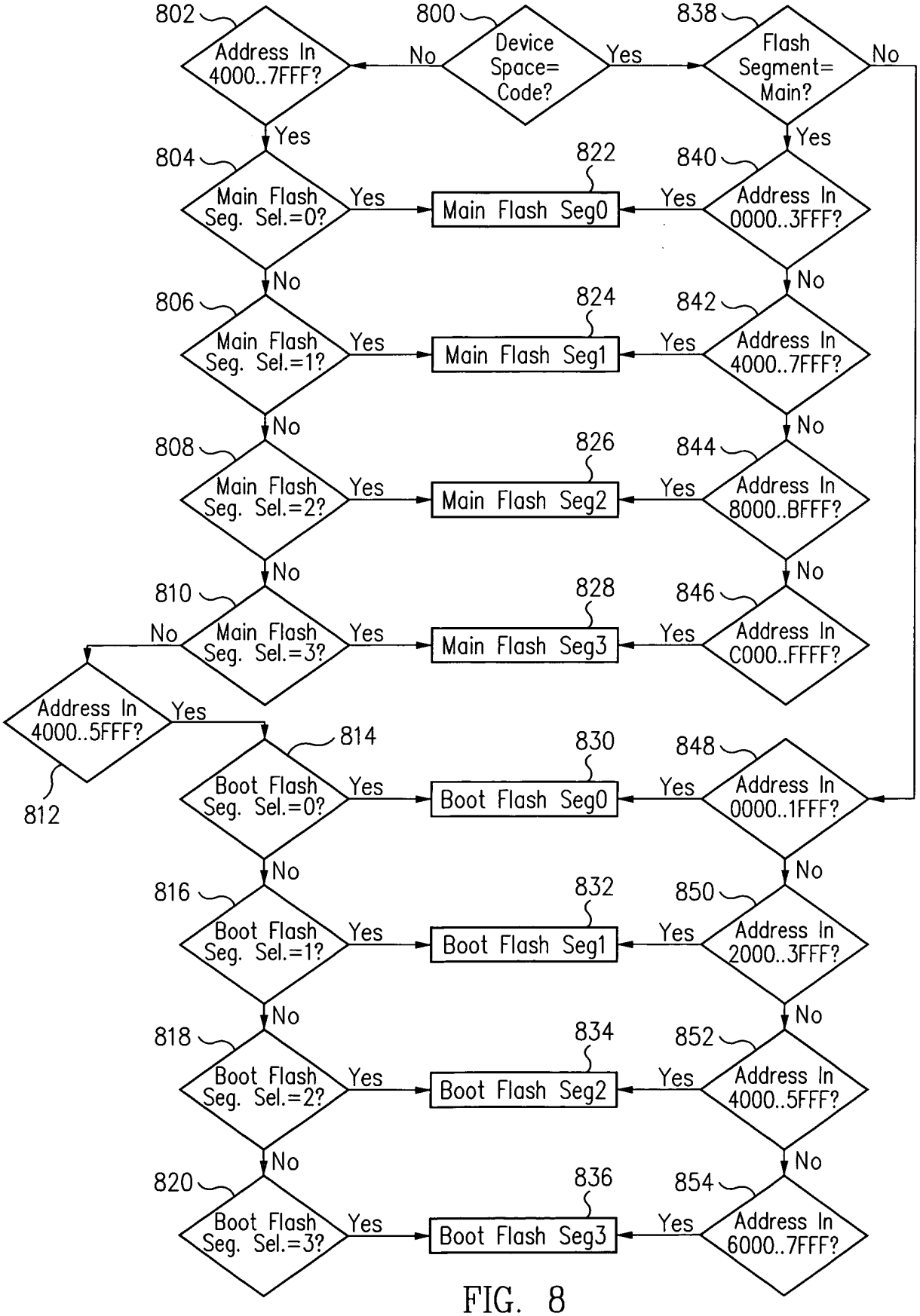
800
Device Space= Code?
No
Yes
802
Address In 4000..7FFF?
838
Flash Segment= Main?
No
Yes
Yes
804
Main Flash Seg. Sel.=0?
Yes
822
Main Flash Seg0
Yes
840
Address In 0000..3FFF?
No
No
806
Main Flash Seg. Sel.=1?
Yes
824
Main Flash Seg1
Yes
842
Address In 4000..7FFF?
No
No
808
Main Flash Seg. Sel.=2?
Yes
826
Main Flash Seg2
Yes
844
Address In 8000..BFFF?
No
No
810
Main Flash Seg. Sel.=3?
No
Yes
828
Main Flash Seg3
Yes
846
Address In C000..FFFF?
Address In 4000..5FFF?
Yes
812
814
Boot Flash Seg. Sel.=0?
Yes
830
Boot Flash Seg0
Yes
848
Address In 0000..1FFF?
No
No
816
Boot Flash Seg. Sel.=1?
Yes
832
Boot Flash Seg1
Yes
850
Address In 2000..3FFF?
No
No
818
Boot Flash Seg. Sel.=2?
Yes
834
Boot Flash Seg2
Yes
852
Address In 4000..5FFF?
No
No
820
Boot Flash Seg. Sel.=3?
Yes
836
Boot Flash Seg3
Yes
854
Address In 6000..7FFF?

FIG. 8

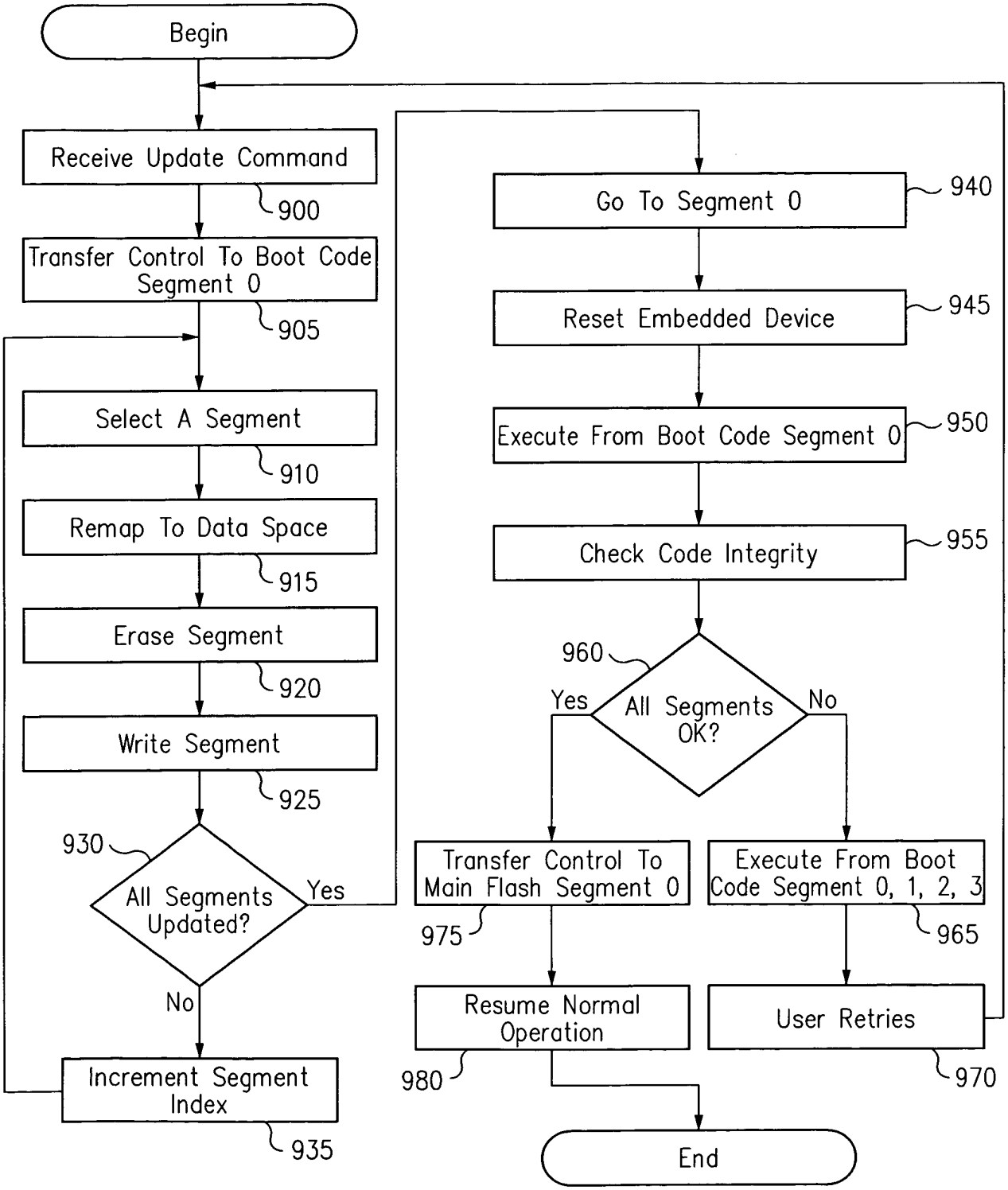
Begin
Receive Update Command
900
Transfer Control To Boot Code Segment 0
905
Select A Segment
910
Remap To Data Space
915
Erase Segment
920
Write Segment
925
930
All Segments Updated?
Yes
No
Increment Segment Index
935
Go To Segment 0
940
Reset Embedded Device
945
Execute From Boot Code Segment 0
950
Check Code Integrity
955
960
All Segments OK?
Yes
No
Transfer Control To Main Flash Segment 0
975
Execute From Boot Code Segment 0, 1, 2, 3
965
Resume Normal Operation
980
User Retries
970
End

FIG. 9

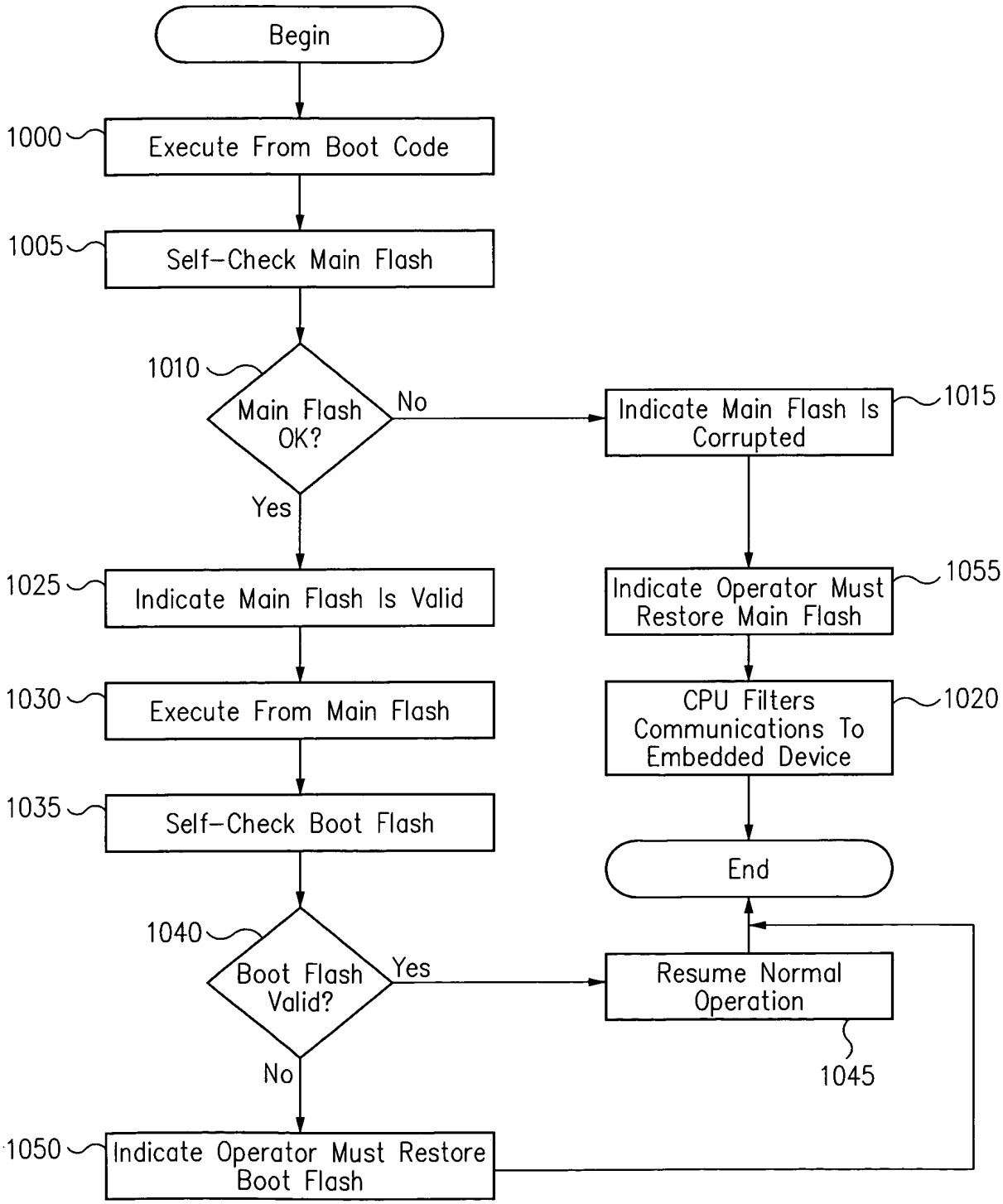
Begin
1000
Execute From Boot Code
1005
Self-Check Main Flash
1010
Main Flash OK?
No
Yes
1015
Indicate Main Flash Is Corrupted
1025
Indicate Main Flash Is Valid
1055
Indicate Operator Must Restore Main Flash
1030
Execute From Main Flash
1020
CPU Filters Communications To Embedded Device
1035
Self-Check Boot Flash
End
1040
Boot Flash Valid?
Yes
Resume Normal Operation
1045
No
1050
Indicate Operator Must Restore Boot Flash

FIG. 10

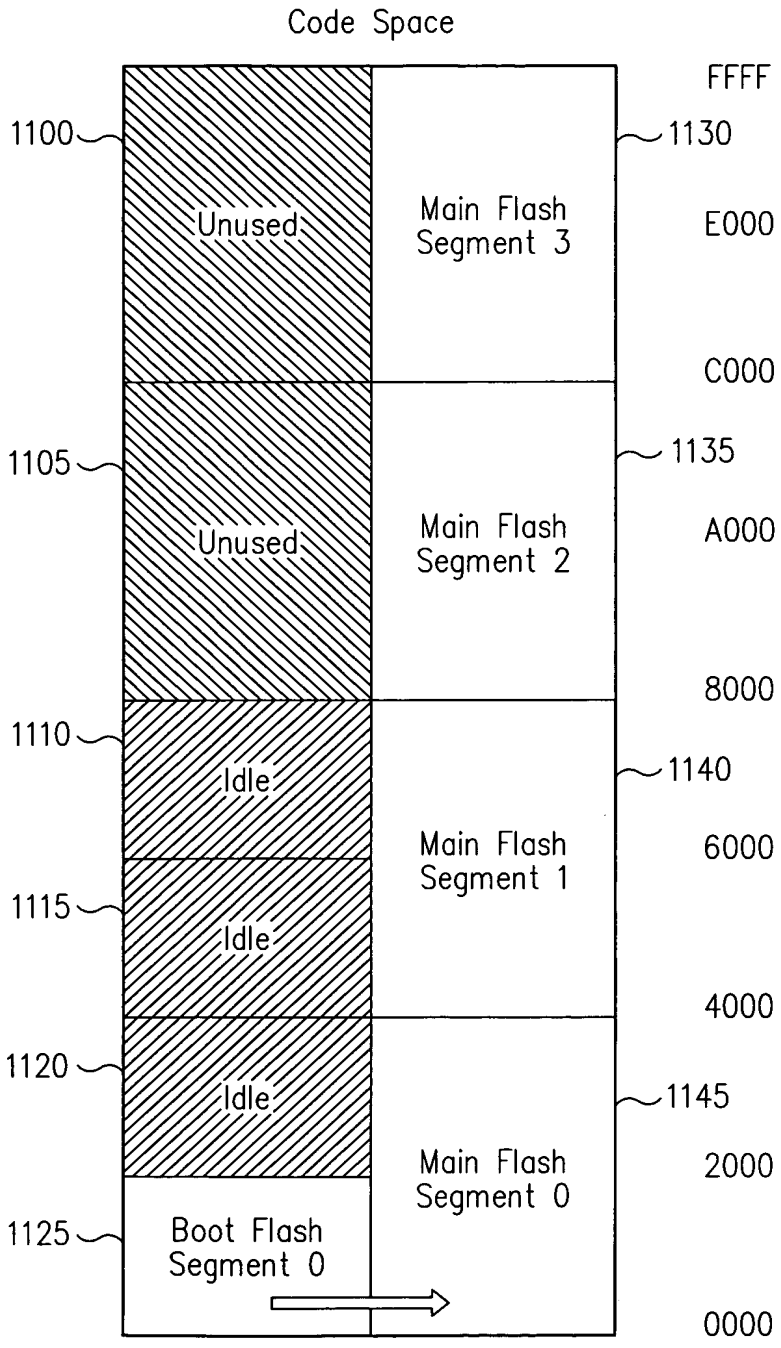
Code Space
FFFF
1100
1130
Unused
Main Flash Segment 3
E000
C000
1105
1135
Unused
Main Flash Segment 2
A000
8000
1110
1140
Idle
Main Flash Segment 1
6000
1115
Idle
4000
1120
1145
Idle
Main Flash Segment 0
2000
1125
Boot Flash Segment 0
0000

FIG. 11

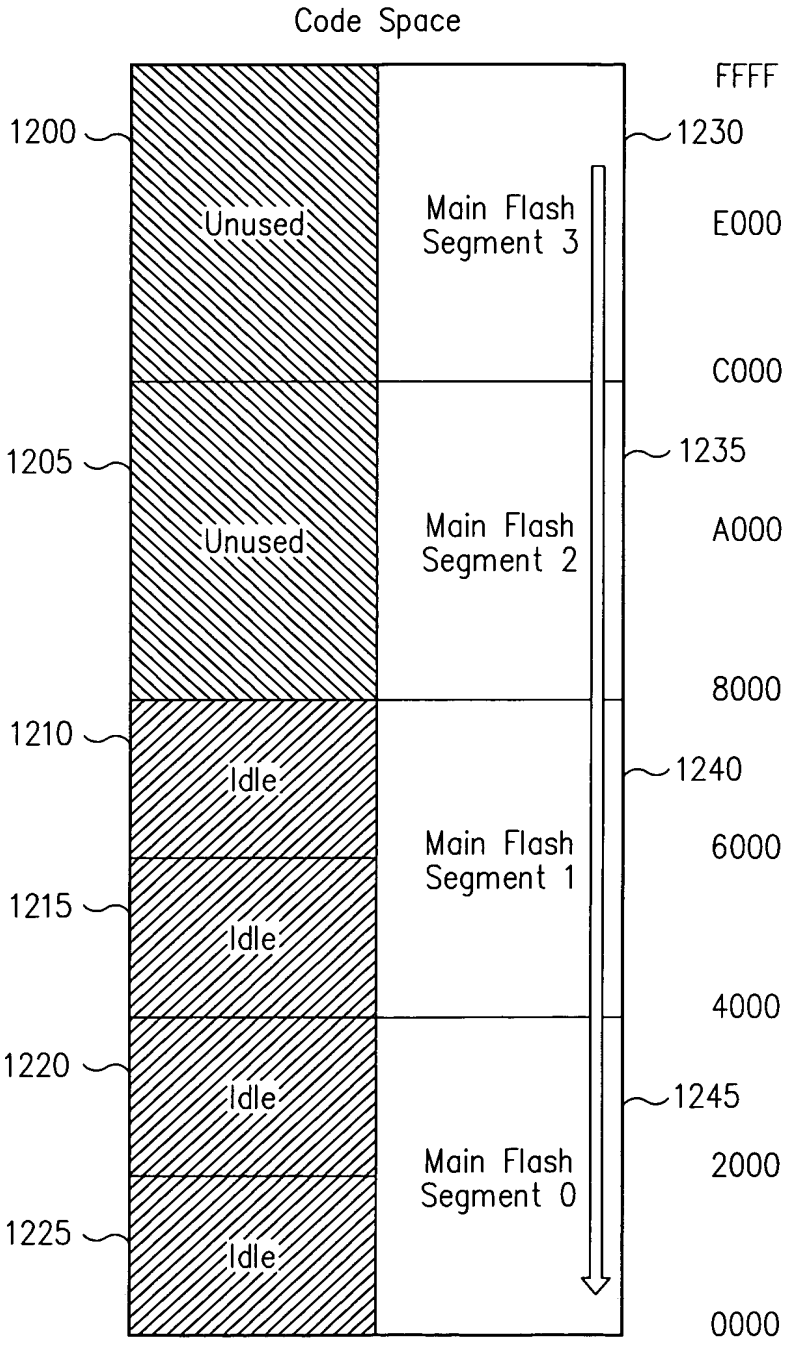
Code Space
FFFF
1200
1230
Unused
Main Flash Segment 3
E000
C000
1205
1235
Unused
Main Flash Segment 2
A000
8000
1210
1240
Idle
Main Flash Segment 1
6000
1215
Idle
4000
1220
1245
Idle
Main Flash Segment 0
2000
1225
Idle
0000

FIG. 12

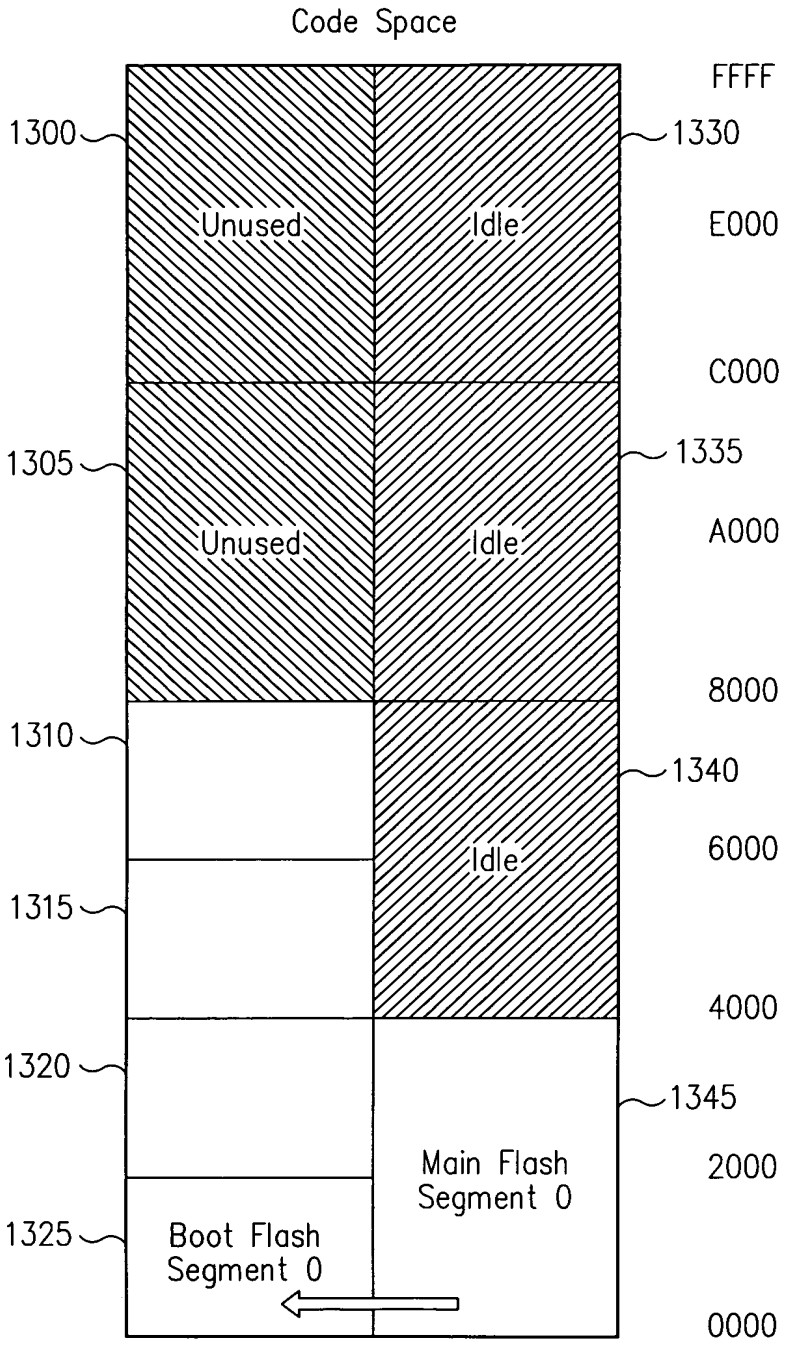
Code Space
FFFF
1300
1330
Unused
Idle
E000
C000
1305
1335
Unused
Idle
A000
8000
1310
1340
Idle
6000
1315
4000
1320
1345
Main Flash
Segment 0
2000
1325
Boot Flash
Segment 0
0000

FIG. 13

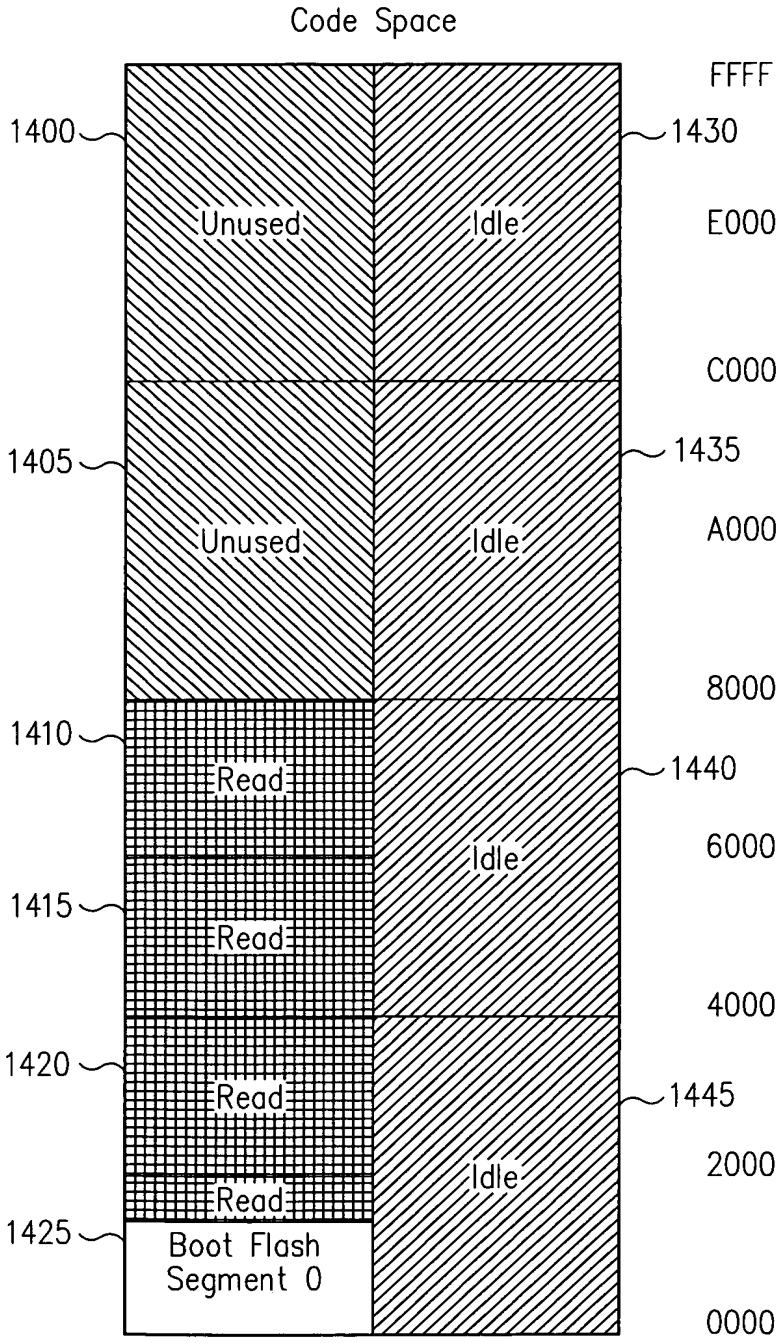
Code Space
FFFF
1400
1430
Unused
Idle
E000
C000
1405
1435
Unused
Idle
A000
8000
1410
1440
Read
Idle
6000
1415
Read
4000
1420
1445
Read
Idle
2000
Read
1425
Boot Flash
Segment 0
0000

FIG. 14

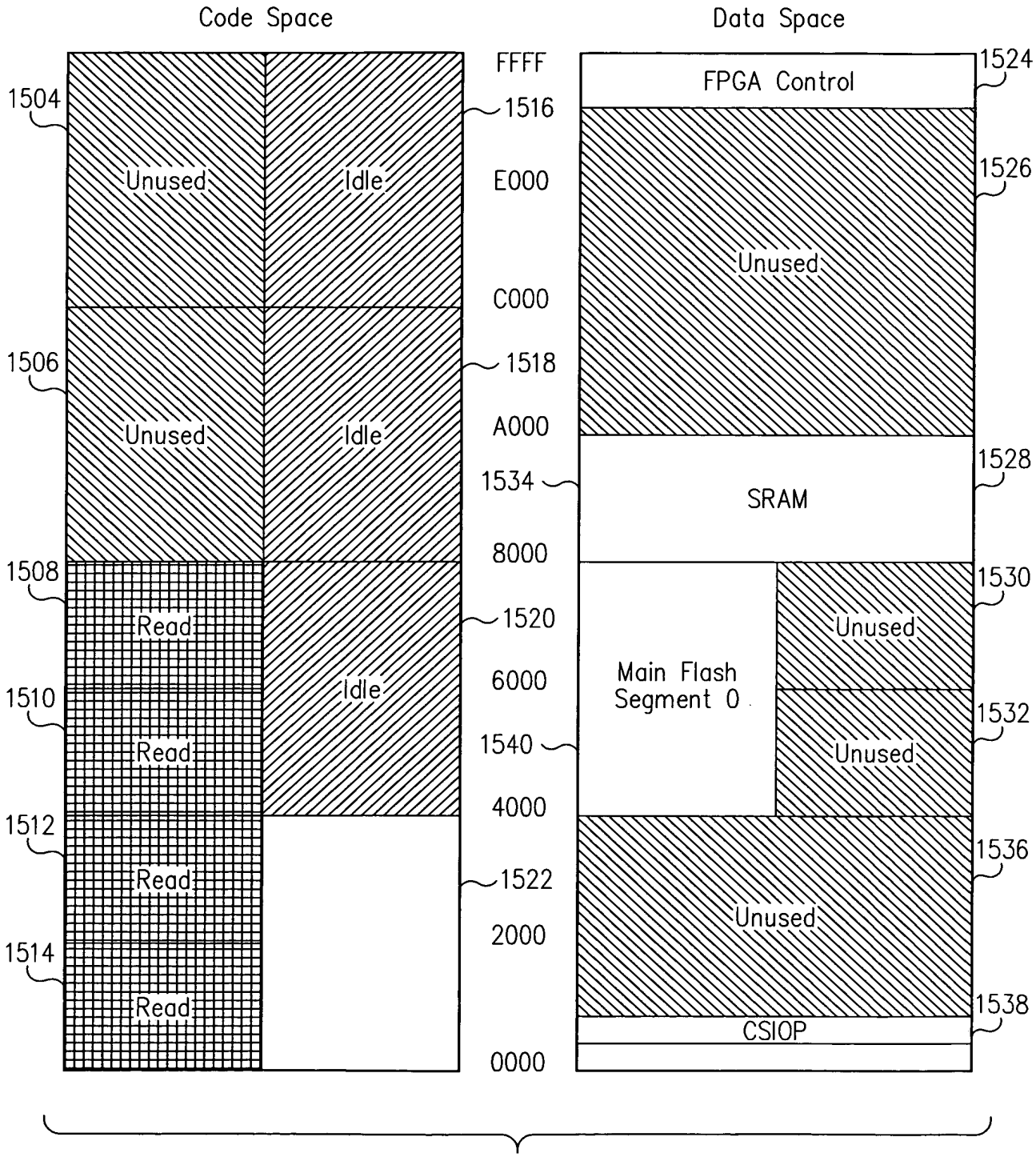
Code Space
Data Space
FFFF
E000
C000
A000
8000
6000
4000
2000
0000
1504
1506
1508
1510
1512
1514
1516
1518
1520
1522
1524
1526
1528
1530
1532
1534
1536
1538
1540
Unused
Idle
Read
FPGA Control
SRAM
Main Flash
Segment 0
CSIOP

FIG. 15

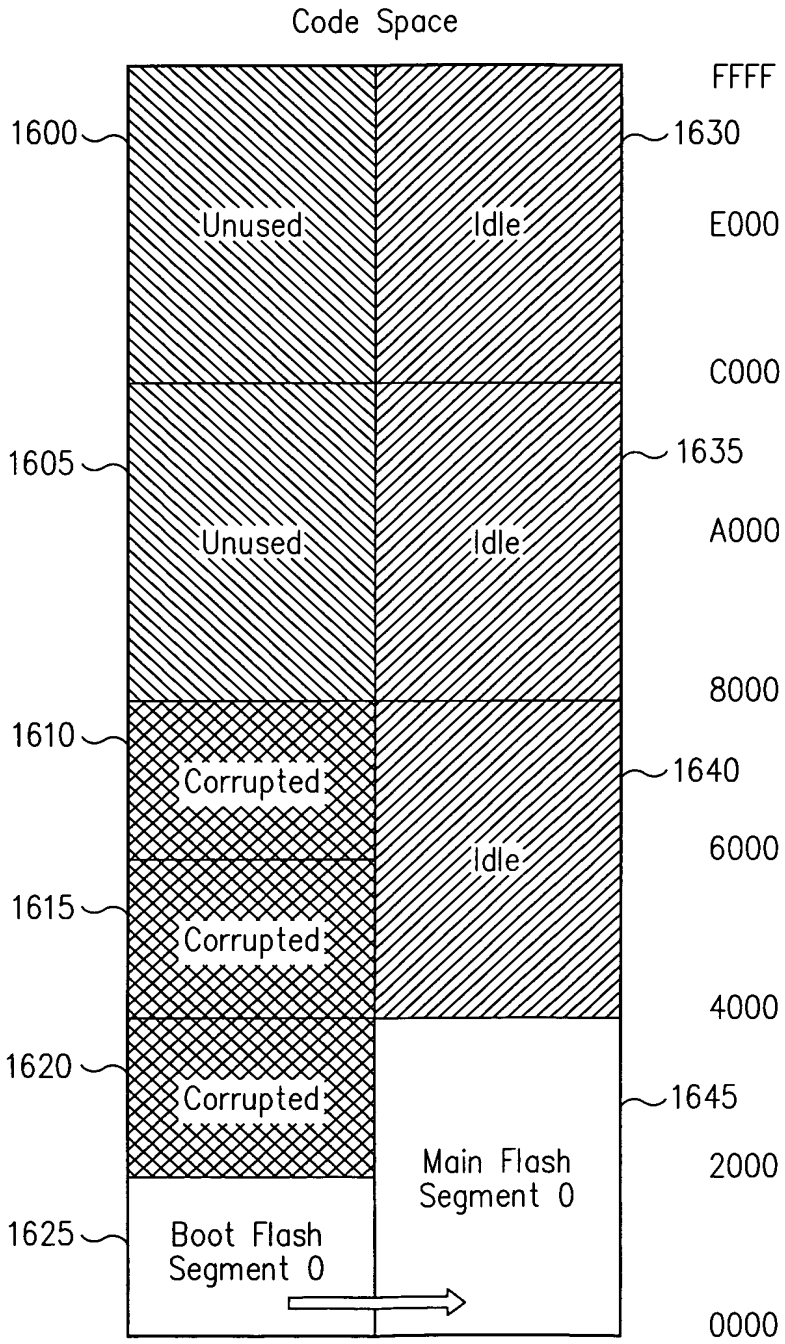
Code Space
FFFF
1600
1630
Unused
Idle
E000
C000
1605
1635
Unused
Idle
A000
8000
1610
1640
Corrupted
Idle
6000
1615
Corrupted
4000
1620
1645
Corrupted
Main Flash
Segment 0
2000
1625
Boot Flash
Segment 0
0000

FIG. 16

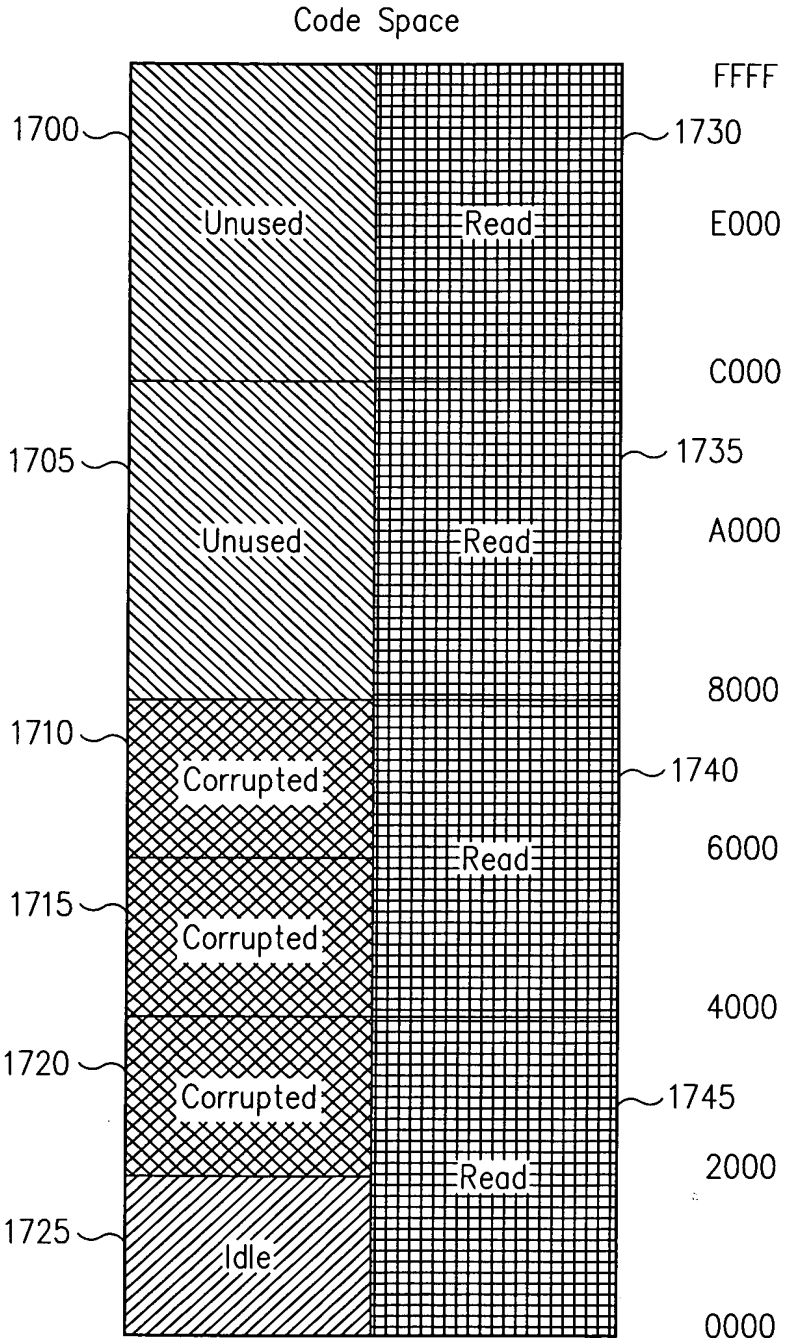
Code Space
FFFF
1700
1730
Unused
Read
E000
C000
1705
1735
Unused
Read
A000
8000
1710
1740
Corrupted
Read
6000
1715
Corrupted
4000
1720
1745
Corrupted
Read
2000
1725
Idle
0000

FIG. 17

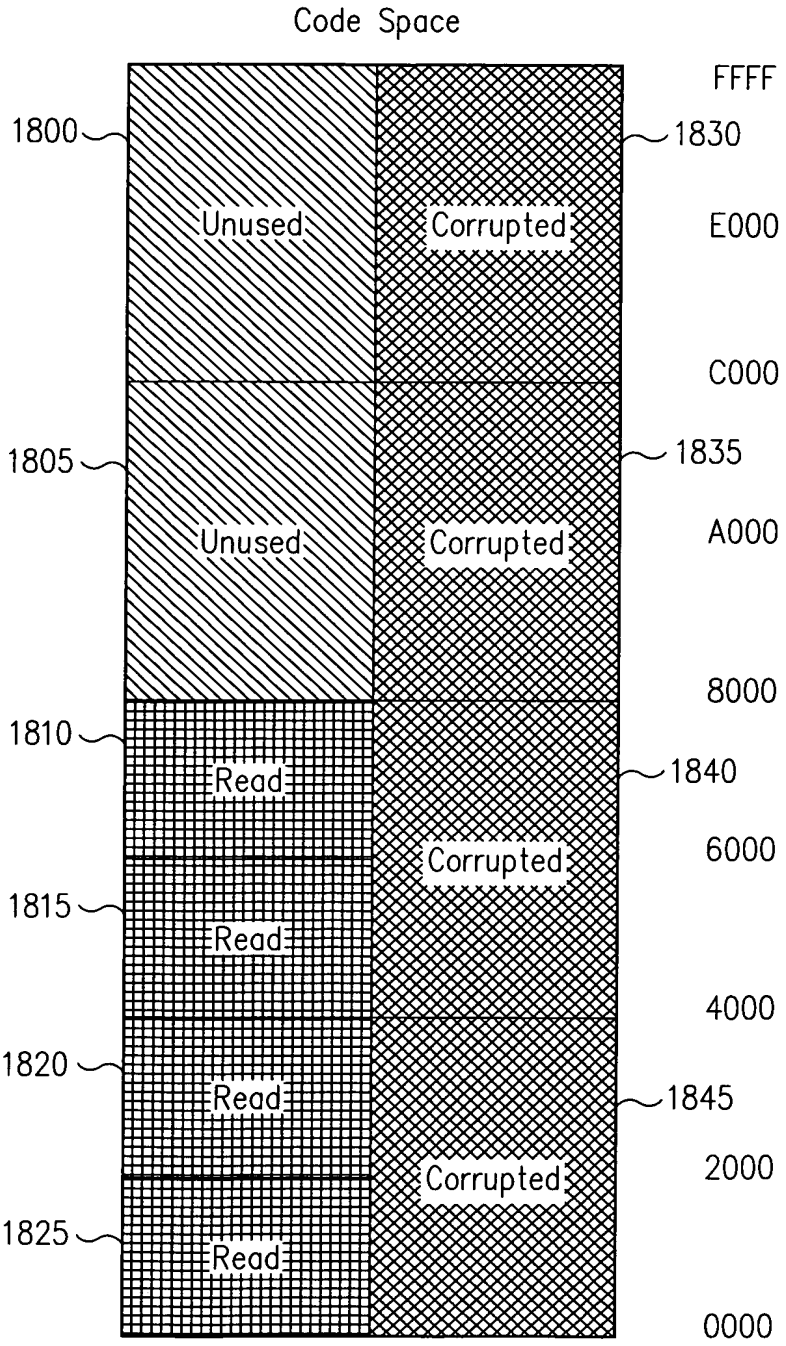
Code Space
FFFF
1800
1830
Unused
Corrupted
E000
C000
1805
1835
Unused
Corrupted
A000
8000
1810
1840
Read
Corrupted
6000
1815
Read
4000
1820
Read
1845
Corrupted
2000
1825
Read
0000

FIG. 18

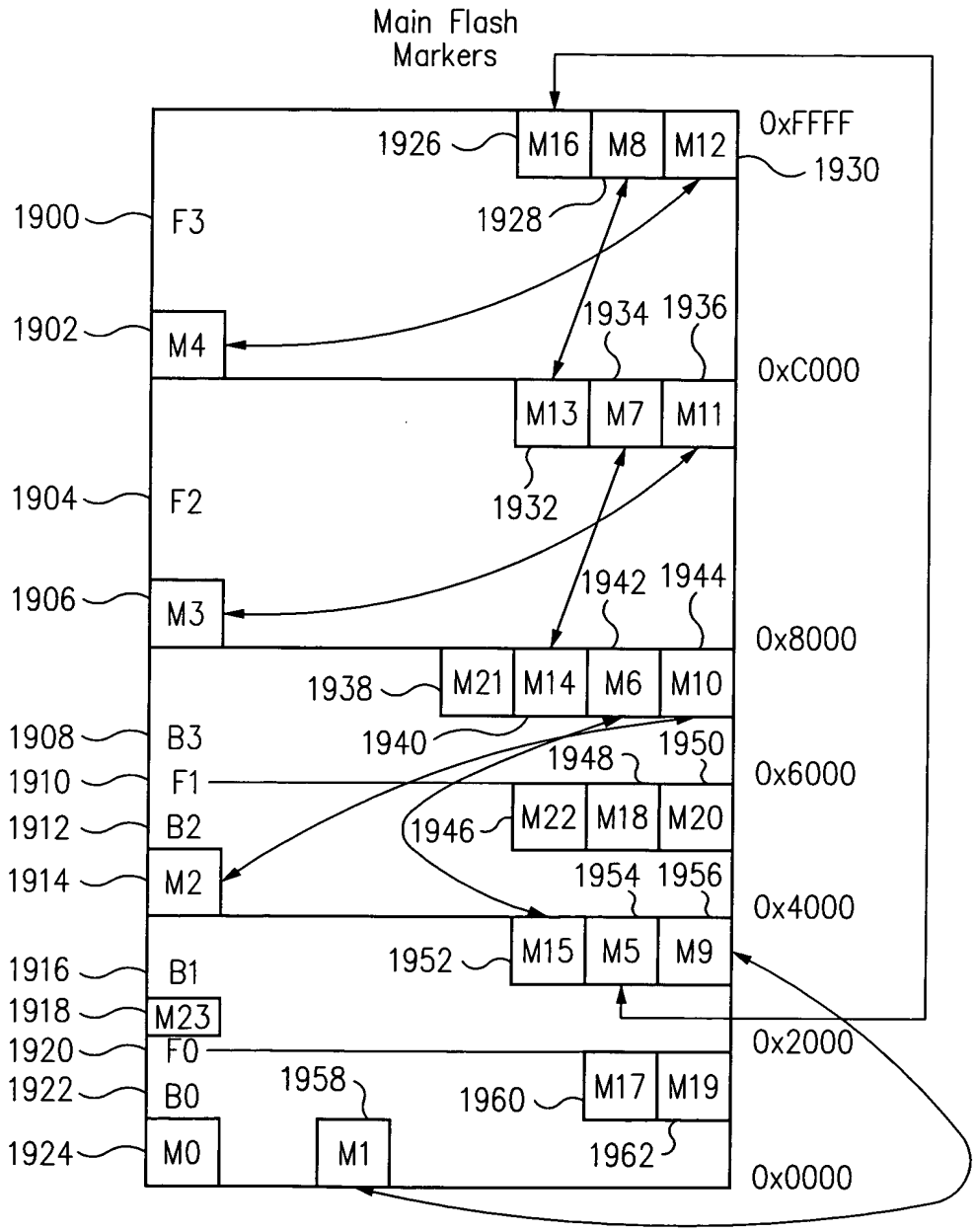
Main Flash
Markers
1926
M16
M8
M12
0xFFFF
1930
1928
1900
F3
1902
M4
1934
1936
0xC000
M13
M7
M11
1932
1904
F2
1906
M3
1942
1944
0x8000
1938
M21
M14
M6
M10
1908
B3
1940
1950
1948
1910
F1
0x6000
1912
B2
1946
M22
M18
M20
1914
M2
1954
1956
0x4000
1916
B1
1952
M15
M5
M9
1918
M23
0x2000
1920
F0
1922
B0
1958
1960
M17
M19
1924
M0
M1
1962
0x0000

FIG. 19

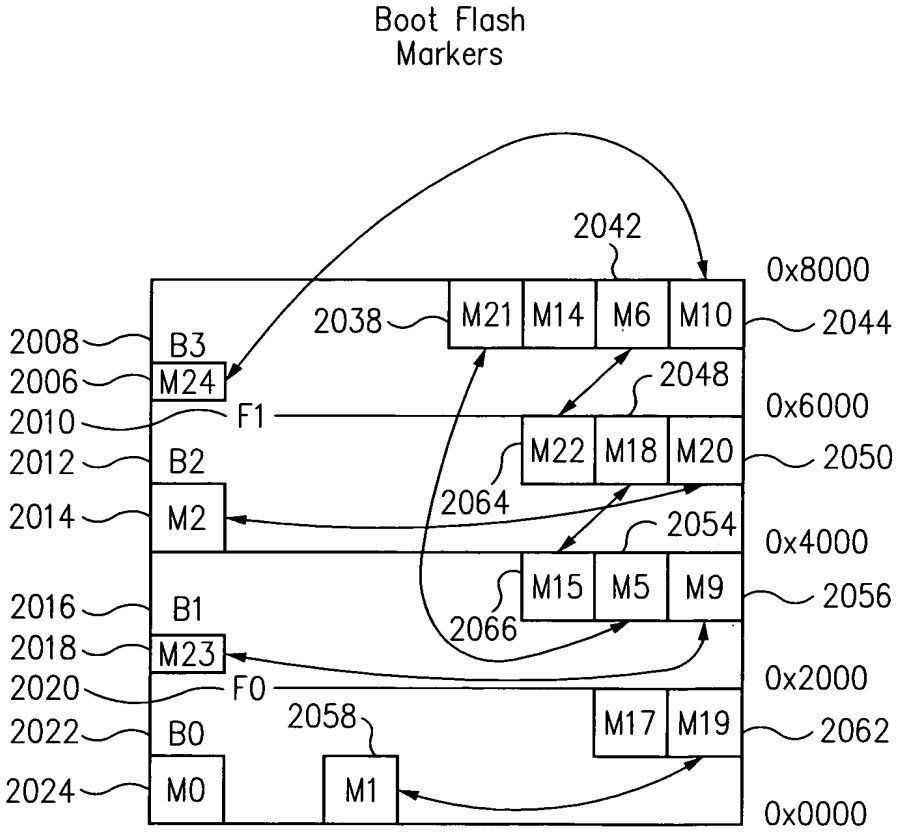
Boot Flash
Markers
2042
0x8000
2038
M21
M14
M6
M10
2044
2008
B3
2006
M24
2048
2010
F1
0x6000
2012
B2
M22
M18
M20
2050
2064
2014
M2
2054
0x4000
M15
M5
M9
2056
2016
B1
2066
2018
M23
0x2000
2020
F0
2058
M17
M19
2062
2022
B0
2024
M0
M1
0x0000

FIG. 20

Marker Correlation

M	~M	
M4	M12	2100
M3	M11	
M2	M10	
M1	M9	
M8	M13	2105
M7	M14	
M6	M15	
M5	M16	

FIG. 21A

M	~M	
M24	M10	2110
M2	M20	
M23	M9	
M1	M19	
M5	M21	2115
M18	M15	
M6	M22	

FIG. 21B

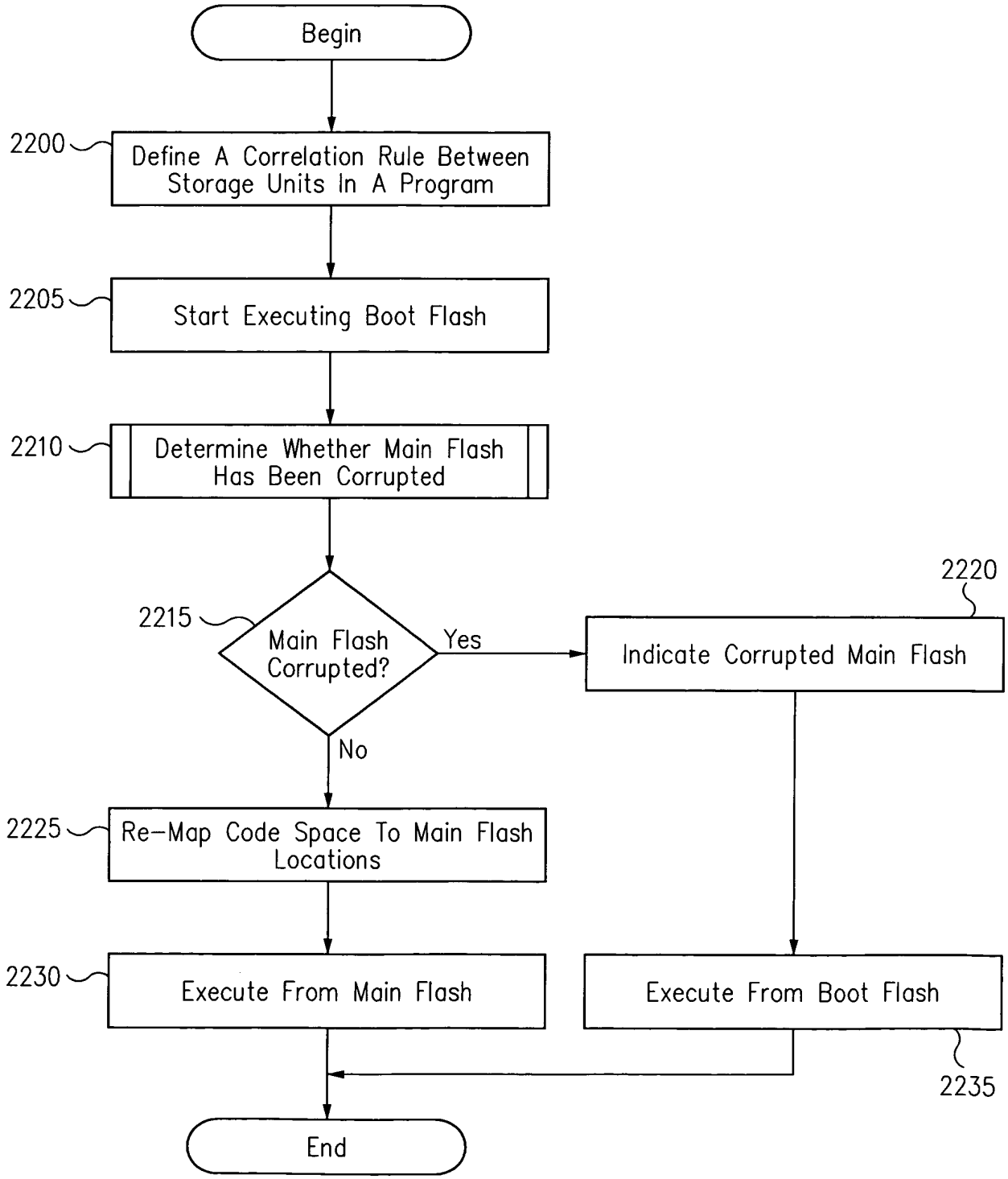
Begin
2200
Define A Correlation Rule Between Storage Units In A Program
2205
Start Executing Boot Flash
2210
Determine Whether Main Flash Has Been Corrupted
2215
Main Flash Corrupted?
Yes
2220
Indicate Corrupted Main Flash
No
2225
Re-Map Code Space To Main Flash Locations
2230
Execute From Main Flash
Execute From Boot Flash
2235
End

FIG. 22

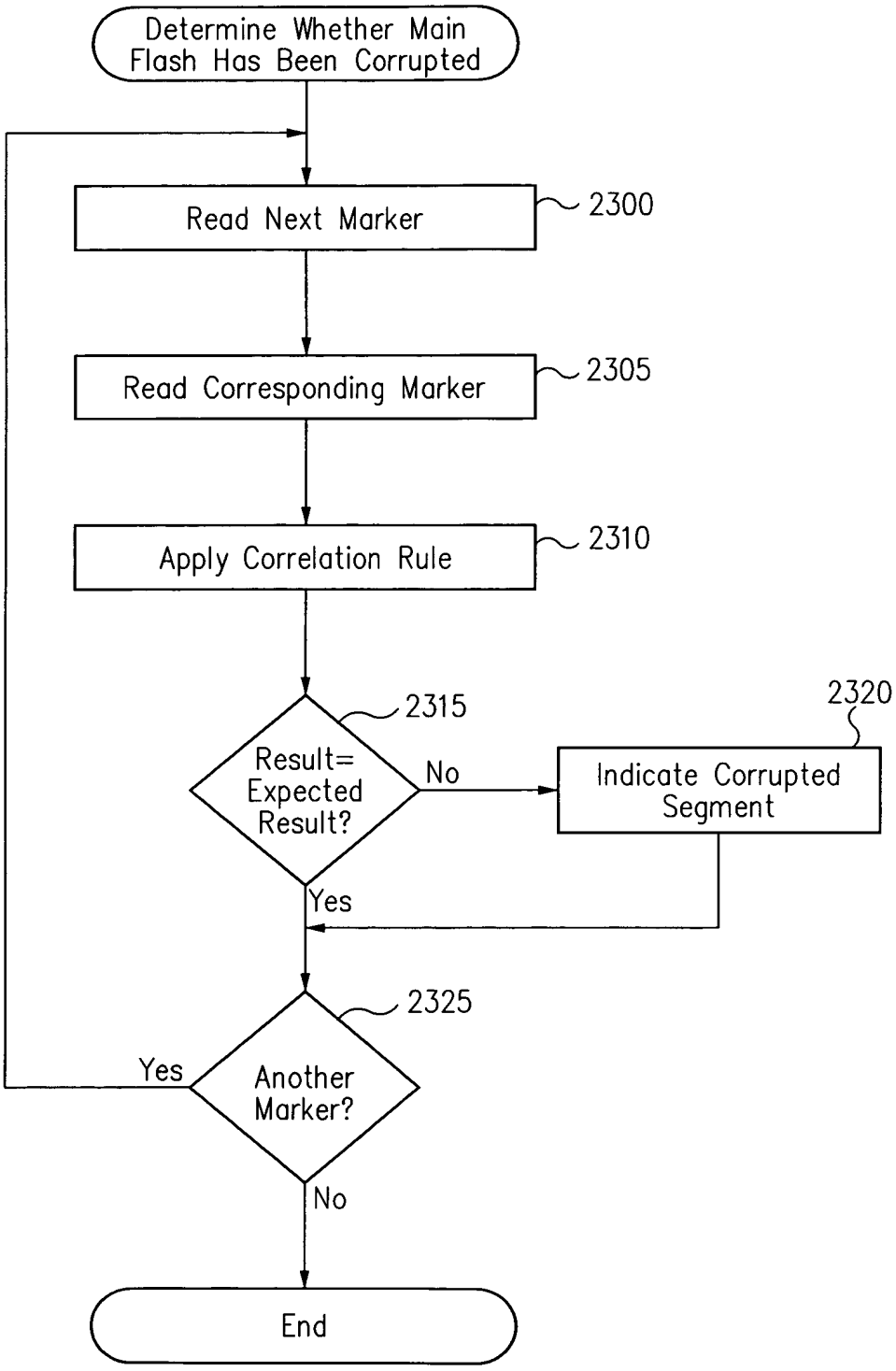
Determine Whether Main Flash Has Been Corrupted
Read Next Marker
2300
Read Corresponding Marker
2305
Apply Correlation Rule
2310
2315
Result= Expected Result?
No
2320
Indicate Corrupted Segment
Yes
2325
Another Marker?
Yes
No
End

FIG. 23